HPC parallelisation of Boundary Conditions in Multiscale Methods

Nikolaos Asproulis and Dimitris Drikakis[1]
Fluid Mechanics and Computational Science,
Cranfield University
Cranfield, Bedfordshire, MK43 0AL
United Kingdom

Abstract This paper investigates two numerical implementations of continuum boundary conditions in parallel high performance computing (HPC) systems in conjunction with multiscale modelling comprising molecular dynamics (MD) and computational fluid dynamics (CFD) methods. The multiscale method provides the best compromise in terms of accuracy and computational cost in mesoscale regimes, however, there are still algorithmic challenges preventing the practical application of these methods. The present study investigates some of these challenges, namely different domain decompositions of the momentum transferred from the continuum domain to the atomistic region in conjunction with HPC parallelisation.

Keywords: hybrid atomistic-continuum, boundary conditions, molecular dynamics, multiscale modelling, parallel computing, computational fluid dynamics

1 Introduction

Miniaturisation of conventional devices offers several advantages, including reduced operating volume, increased throughput and improved accuracy [1–5]. As the operational dimensions reduce to smaller scales the surface-to-volume ratio increases and the phenomena observed are characterised by an inherent multiscale nature [6–8]. High-fidelity computational modelling of micro and nano fluid flows based on the Euler/Navier-Stokes equations (for the continuum scales) and molecular dynamics (for the atomistic scales) increasingly receives more attention amongst academic research groups because of its promising computational features with respect to the design and optimisation of micro/nano devices [9–12].

The traditional continuum models tend to neglect the microscopic mechanisms of the phenomena involved in these scales and, therefore, cannot

[1]Corresponding author, d.drikakis@cranfield.ac.uk

entirely represent the fluid flow physics inside micro and nano scale systems [13]. In the cases where the macroscopic constitutive relations or boundary conditions become inadequate, microscopic models such as molecular dynamics (MD) have to be employed. The Achilles heel of the molecular simulations is the high computational cost that restricts their applications to nanoscale systems and time scales below microseconds. To circumvent the implications arising from the disparity of scales, both spatial and temporal multiscale frameworks have been developed [2, 11, 14–20]. The accuracy and efficiency of the above methods is based on the information exchange between the continuum and atomistic regions. A challenging task is to impose macroscopic boundary conditions on the molecular domain because the microscopic description is associated with more degrees of freedom than the macroscopic one.

This paper presents two parallel implementations of the pressure across the continuum-atomistic interface and discusses the associated parallel performance issues.

2 Methodology

In hybrid continuum (CFD)-molecular (MD) modelling, consistency of the physical (conservation) laws across boundaries of the MD and CFD domains needs to be imposed. The molecular system is, therefore, initialised according to the continuum density, ρ_{con}, and temperature, T_{con}. In the atomistic simulations, the molecular system is constrained through the continuum velocity and temperature gradient by controlling the molecular motion within confined regions of the molecular boundaries. Enforcing the continuum constraints requires altering the properties of the atoms at the boundaries in order to match the continuum velocity, u_{con}, and temperature, T_{con}. Additionally, the local continuum pressure, P_{con}, is applied normal to the outer surface of the constrained region in order to keep the atoms within the molecular domain, as well as to impose the correct pressure onto the molecular system. Figure 1 shows the decomposition of the computational domain to atomistic and continuum regions. The blue boundary cells are used for transferring information from the continuum to the molecular region and the ghost cells for the reverse procedure.

The average velocity of particles in the constrained region R_{BCT} (boundary cells), shown in blue colour in Fig. 1, should correspond to the continuum velocity $\boldsymbol{u}_{con}$. Assuming that N particles with total mass $M = \sum_{i=1}^{i=N} m_i$ and velocities $\boldsymbol{u}_i$ belong to R_{BCT}, the particles velocities should

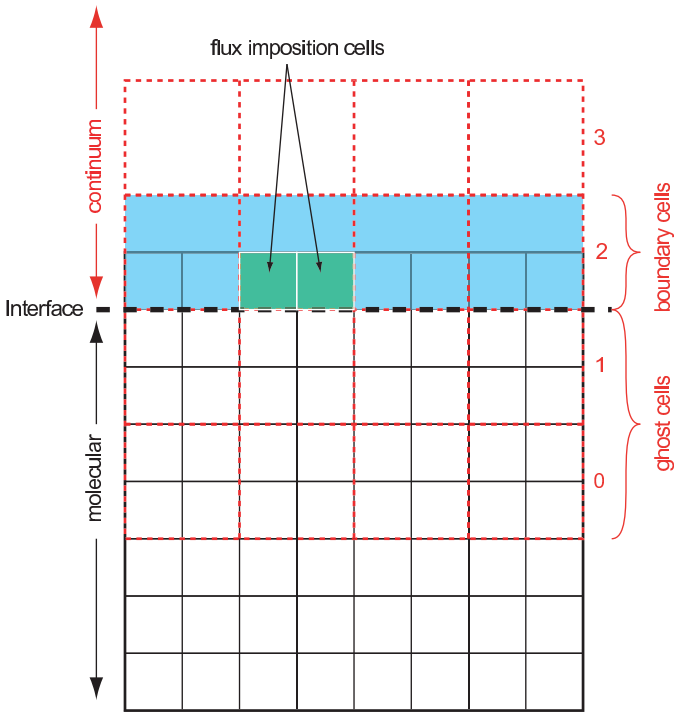


Figure 1: Schematic representation of the domain decomposition in hybrid atomistic-continuum simulations

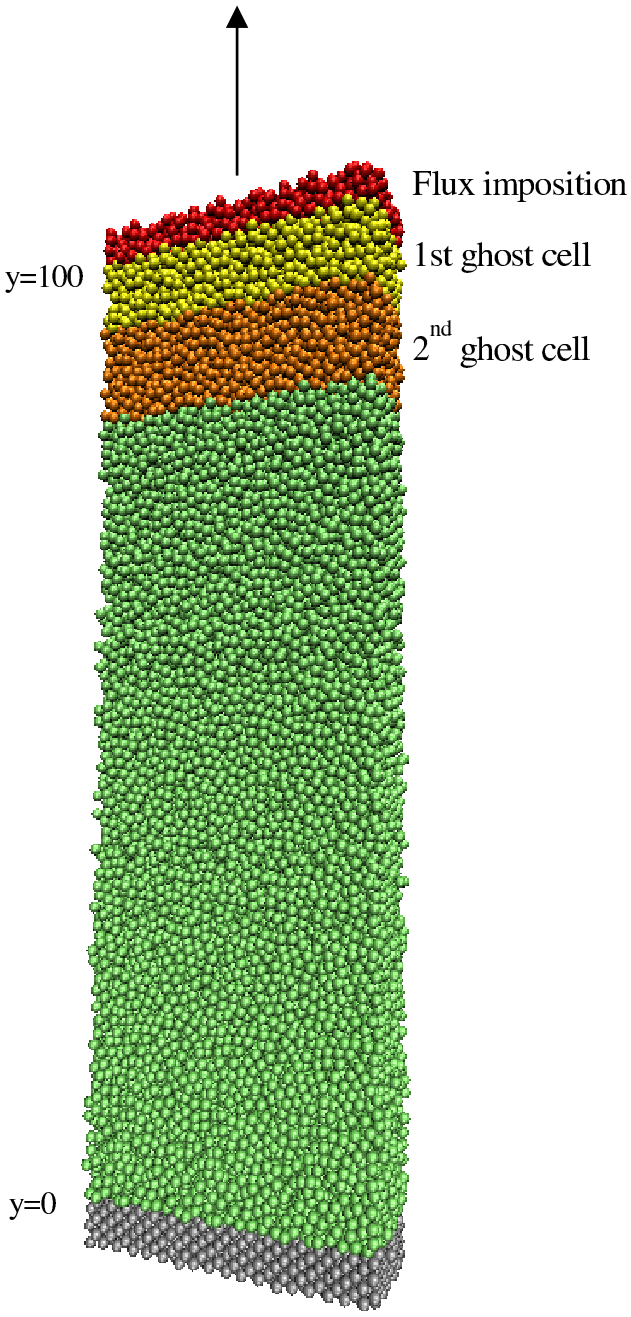


Figure 2: Simulation Set Up

be re-scaled:

$$\boldsymbol{u}_i^{'} = \boldsymbol{u}_i - \frac{1}{M}\sum_{i=1}^{i=N} m_i \boldsymbol{u}_i + \boldsymbol{u}_{con} \ . \tag{1}$$

The continuum temperature is applied to the microscopic system through an energy transfer scheme [2] according to which energy is added to or removed from the atomistic system, so that the atomistic temperature equals the macroscopic one without needing to modify the particles mean velocity. Hybrid methods often apply the normal pressure through external forces [6]. This approach, however, inserts/removes energy depending on the velocity of the atoms subjected to the force. This results in oscillations in the molecular system [2]. The oscillations can be significantly reduced by using the velocity reversing scheme [2], where the pressure, P_{con}, is applied by reversing the velocity vector of atoms that move in the opposite direction of the pressure force. If the outer surface of the constrained region is normal to a dimension α, then an atom i is reversed by changing the sign of the respective velocity component: $\boldsymbol{v}_{i,\alpha}^{'} = -\boldsymbol{v}_{i,\alpha}^{'}$. For each reversed atom, i, a momentum $p_i = 2m_i\boldsymbol{v}_{i,\alpha}^{'}$ is implemented. To apply a pressure P_{con} at each MD time step, the algorithm continues to reverse atoms until the transferred momentum equals the required momentum transfer due to the pressure:

$$\sum_i 2m_i\boldsymbol{v}_i^{'} = P_{con}\Delta t A_{ctr} \ , \tag{2}$$

where the sum is over the reversed atoms, Δt is the time step, and A_{ctr} is the surface area of the constrained region.

The most computationally intensive operation of the hybrid boundary conditions implementation is the velocity reversing scheme for applying the continuum pressure. Two techniques have been developed to parallelise the velocity reversing scheme as shown in Figure 3:

- A-Method: A continuum pressure, P_{con}, corresponding to momentum transfer p_a, is applied to the molecular region, and the total continuum momentum is equally distributed (p_a/N_1) across the N_1 processors, which comprise the first layer of the parallel grid corresponding to the upper boundary transfer region. In this case, each processor has to identify its outermost atoms and reverse their velocities until a total momentum p_a/N_1 is applied. If the upper layer of processors does not contain enough particles for applying the continuum momentum, then the information regarding the remaining momentum is

transferred to the neighbouring lower processor where the same procedure is applied. Using this approach the momentum is distributed and transferred through the individual columns of the parallel grid.

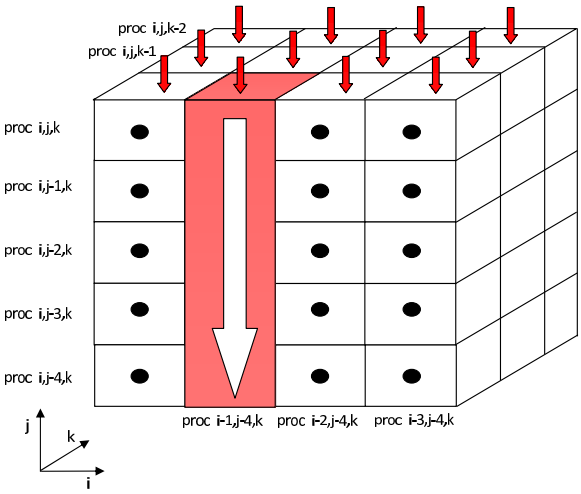
- B-Method: An alternative approach is based on the communication between all allocated processors within the computational domain, aiming at identifing the outermost atoms and apply the velocity reversing scheme directly. Figure 3(b) shows the entire continuum momentum being applied directly to the molecular region and not distributed across the parallel grid, as in Figure 3(a). The main advantage of this method is its physical consistency since the continuum momentum is applied only to the globally outermost atoms that are likely to escape from the atomistic region. Its efficiency depends on the values of the applied continuum pressure.

The key factor that affects the performance efficiency is the searching algorithm for the outermost atoms. The B-Method finds the global outermost atoms before applying the velocity reversing scheme, thus leading to additional communication between the processors at the (i,k) level, as shown in Figure 1. On the other hand, the A-Method inherently assumes a uniform pressure distribution at the molecular-continuum interface. The A-Method calculates the total momentum by the velocity reversing scheme and applies it independently to each (i,j) processor column, as shown in Figure 1. It also performs a local search of the outmost atoms in each processor column until the global continuum pressure is achieved.

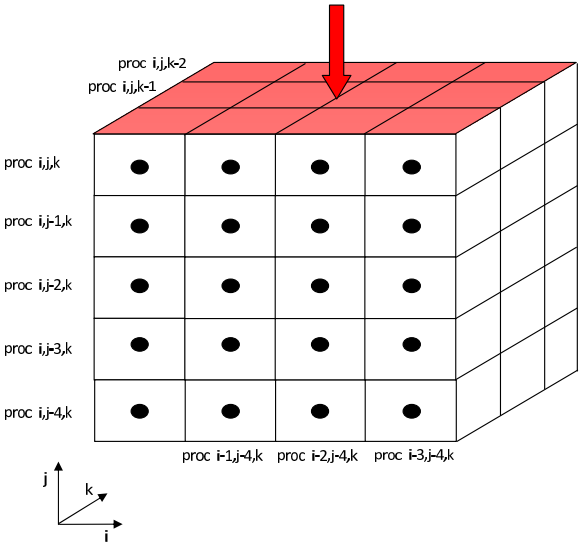
3 Results and Discussion

The parallel implementation of the boundary conditions has been made in conjunction with the LAMMPS molecular dynamics code [21] and its extensions for hybrid atomic-continuum interfaces developed by the authors [2,11,20]. Similar to most MD codes, LAMMPS employs a Cartesian domain decomposition allowing particles to travel from one processor to the other and, therefore, in the execution face it can easily cope with curved tubes and corrugated surfaces (see [11] and references therein). The main limiting factor is associated with the construction of the system and the allocation of the particles to the various processors, especially in the cases where complex potentials are employed; this may lead to load imbalance.

In the present study, the size of the molecular domain is 20σ, 55σ and 20σ in the x, y and z directions, respectively. Periodic boundary conditions



(a) A-Method



(b) B-Method

Figure 3: Parallel implementations of momentum transfer from the continuum to the molecular region

are applied in the x and z directions. Along the y direction the two sides of the domain are constrained by a solid thermal wall and a continuum-based boundary condition. In the present simulations, the thermal wall is modelled by two planes of a (111) face-centred cubic (fcc) lattice with the temperature of the wall being maintained by a velocity rescaling algorithm that is applied to each plane separately. This type of thermal walls operates as heat baths and aims to maintain a thermal equilibrium without the need of an additional thermostat. In the present simulations the density of the wall atoms and the interaction parameters are $\rho_{wall} = 1.0\ m\sigma^{-3}$, $\epsilon_{wf} = 0.6\ \epsilon$ and $\sigma_{wf} = 1.0\ \sigma$. These parameters represent a solid wall with no slip boundary conditions and correspond to a total number of 676 wall particles. The density of the fluid has been set as $\rho_{fluid} = 0.8\ m\sigma^{-3}$ corresponding to the generation of $18,144$ particles. The continuum boundary conditions are applied in $45\sigma < y < 55\sigma$ and the flow region is between $0\sigma < y < 45\sigma$. The simulations are performed for a total 6×10^6 time steps, with $\Delta t_{MD} = 0.001\tau$.

Figure 4 shows the parallel speed up of the molecular code for three different values of continuum pressure. A super-linear speed up is observed due to the increased memory requirements of the serially executed molecular solver. If a smaller system had been employed the miss-cache problems of the serial code might have been avoided but in this case the size of the system would not be appropriate for a hybrid multiscale simulation. The results reveal that for smaller values of continuum pressure, the B-Method presents higher speed-up values, however, as the pressure increases from $P_{con} = 6.32\epsilon\sigma^{-3}$ to $P_{con} = 8.0\epsilon\sigma^{-3}$ the A-Method becomes more efficient. Increased values of pressure correspond to an increased number of particles participating in the velocity reversing scheme, therefore contributing to the reduced efficiency of the B-Method that detects the global outermost particles.

In the case where the calculated pressure values are smaller than the continuum pressure, the algorithm reduces the number of particles for the velocity reversal, making it comparable to the actual number of processors allocated to the top (i,k) level. As a result, it becomes more efficient to globally search (B-Method) for the few outermost atoms and reverse their velocities rather than allocating the pressure values to the processor columns (A-Method). As the values of pressure increase, the number of outermost atoms also increases, thus penalising the *global search* approach and favouring the A-Method that distributes the workload to the processor columns. The global search approach (B-Method) utilises MPI collective communication commands and its performance is better than the B-Method only in the

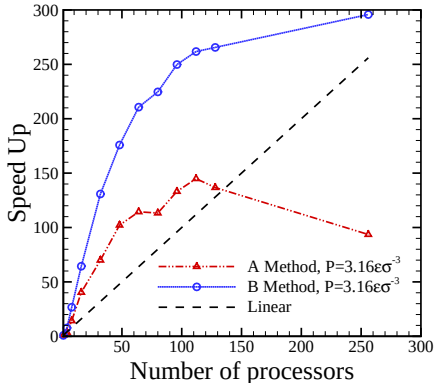
cases that the number of particles that needs to be detected is smaller than the number of processors allocated to the first Cartesian level (i,k). This is shown in Figure 4(a) and explains the improved speedup when increasing the total number of processors.

When increasing the number of processors allocated to the top level, the comparative performance of the B- over A-Method also increases. However, as the continuum pressure values increase leading to numbers of particles comparable to the number of processors (Figure 4(b) for number of processors less than 100) the performance of the two methods is at the same level. As the number of processors increases to 256, the B-Method starts showing a performance advantage. If the pressure continues to increase even further (Figure 4(c)) then the number of particles that need to be identified significantly increases, hence the A-Method becomes more efficient. However, as the number of processors further increases this advantage no longer exists and the total simulation time for 256 processors remains the same for both methods, as shown in Fig. 4(c).

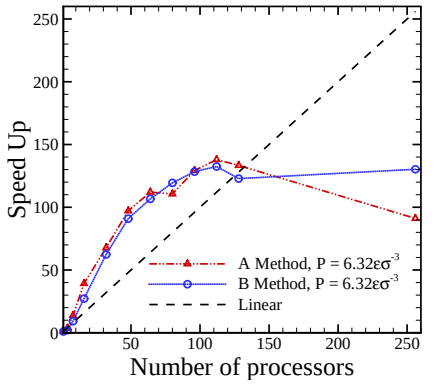
Figure 5 shows the total simulation time as a function of pressure, for both the A- and B-Method, as well as for a number of processors. The time required for a simulation using the B-Method increases non-linearly with pressure. On the other hand, the A-Method which distributes the total pressure to the number of processors of the top layer, presents almost a constant performance as a function of pressure. For lower pressure values ($P_{con} = 1.58\epsilon\sigma^{-3}$), the B-Method is faster even on 64 processors compared to the A-Method on 128 processors.

4 Concluding remarks

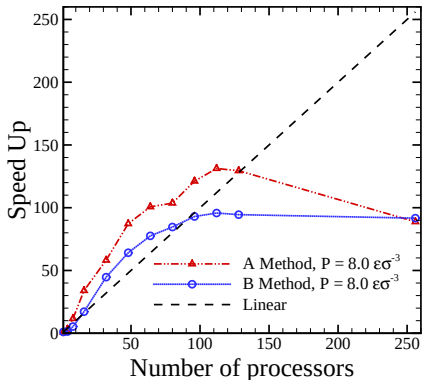
The present study shows that the method used for transferring boundary conditions in the parallel implementation of a multi-scale hybrid CFD-MD method can significantly affect the performance of the computational algorithm. The overall performance depends on the simulation problem, number of processor used, total number of particles and continuum pressure applied. The main advantage of the A-Method is consistent performance when increasing the number of processors and better efficiency than the B-Method at higher pressure values, whereas the B-Method shows better performance at lower pressure values. Further work needs to be performed to avoid cache and memory problems encountered in connection with the sequential execution of multiscale codes. Finally, the implementation of the above techniques in conjunction with incompressible and compressible CFD methods devel-



(a) $P_{con} = 3.16\epsilon\sigma^{-3}$



(b) $P_{con} = 6.32\epsilon\sigma^{-3}$



(c) $P_{con} = 8.0\epsilon\sigma^{-3}$

Figure 4: Parallel speed up for various values of continuum pressure

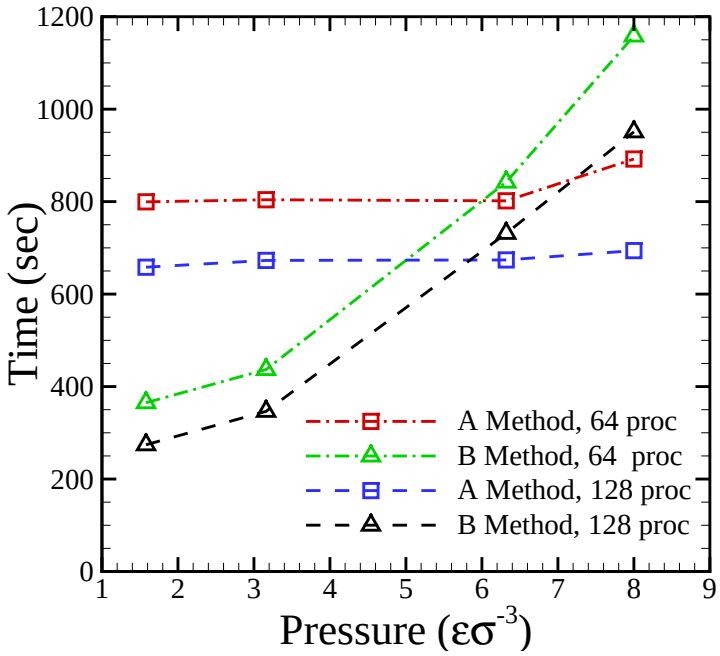


Figure 5: Total simulation time as function of pressure for boundary condition techniques labelled as A- and B-Method, and for a varying number of processors

oped by the authors [22–25] is underway.

References

[1] W. D. Nicholls, M. K. Borg, D. A. Lockerby, and J. M. Reese. Water transport through carbon nanotubes with defects. *Molecular Simulation*, 38(10):781–785, 2012.

[2] M Kalweit and D Drikakis. Coupling strategies for hybrid molecular-continuum simulation methods. *Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science*, 222:797–806(10), 2008.

[3] M. Kalweit and D. Drikakis. Multiscale methods for micro/nano flows and materials. *J. Comput. Theo. Nano Sci.*, 5(9):1923–1938, 2008.

[4] N. Asproulis and D. Drikakis. Nanoscale materials modelling using neural networks. *J. Comput. Theo. Nano Sci.*, 6(3):514–518, 2009.

[5] F. D. Sofos, T. E. Karakasidis, and A. Liakopoulos. Effects of wall roughness on flow in nanochannels. *Phys. Rev. E*, 79(2), 2009.

[6] Thomas Werder, Jens Walther, and Petros Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flows. *J. Comput. Phys.*, 205:373–390, 2005.

[7] F. Sofos, T. E. Karakasidis, and A. Liakopoulos. Surface wettability effects on flow in rough wall nanochannels. *Microfluidics and Nanofluidics*, 12(1-4):25–31, 2012.

[8] A. E. Giannakopoulos, F. Sofos, T. E. Karakasidis, and A. Liakopoulos. Unified description of size effects of transport properties of liquids flowing in nanochannels. *International Journal of Heat and Mass Transfer*, 55(19-20):5087–5092, 2012.

[9] N. V. Priezjev. Molecular diffusion and slip boundary conditions at smooth surfaces with periodic and random nanoscale textures. *Journal of Chemical Physics*, 135(20), 2011.

[10] N. V. Priezjev. Interfacial friction between semiflexible polymers and crystalline surfaces. *Journal of Chemical Physics*, 136(22), 2012.

[11] N. Asproulis and D. Drikakis. Surface roughness effects in micro and nanofluidic devices. *J. Comput. Theo. Nano Sci.*, 7(9):1825–1830, 2010.

[12] M. Benke, E. Shapiro, and D. Drikakis. An efficient multi-scale modelling approach for ssdna motion in fluid flow. *J. Bio. Eng.*, 5(4):299–307, 2008.

[13] J. Liu, S. Chen, X. Nie, and M. O. Robbins. A continuum-atomistic simulation of heat transfer in micro- and nano-flows. *J. Comput. Phys.*, 227(1):279–291, 2007.

[14] R. Delgado-Buscalioni and P.V. Coveney. Continuum-particle hybrid coupling for mass, momentum and energy transfers. *Phys. Rev. E*, 67:046704, 2003.

[15] N. G. Hadjiconstantinou. Combining atomistic and continuum simulations of contact-line motion. *Phys. Rev. E*, 59:2475, 1999.

[16] W. Ren and E. Weinan. Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics. *J. Comput. Phys.*, 204(1):1–26, 2005.

[17] M. K. Borg, D. A. Lockerby, and J. M. Reese. Fluid simulations with atomistic resolution: a hybrid multiscale method with field-wise coupling. *Journal of Computational Physics*, 255:149–165, 2013.

[18] C. J. Garcia-Cervera, W. Ren, J. Lu, and E. Weinan. Sequential multiscale modeling using sparse representation. *Comm. Comp. Ph.*, 4(5):1025–1033, 2008.

[19] E. Weinan. The heterogeneous multi-scale method: A mathematical framework for multi-scale modeling, 2005.

[20] N. Asproulis, M. Kalweit, and D. Drikakis. A hybrid molecular continuum method using point wise coupling. *Advances in Engineering Software*, 46(1):85–92, 2012.

[21] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, 1995.

[22] Evgeniy Shapiro and Dimitris Drikakis. Artificial compressibility, characteristics-based schemes for variable density, incompressible, multi-species flows: part i. derivation of different formulations and constant density limit. *J. Comput. Phys.*, 210(2):584–607, 2005.

[23] D. Drikakis, M. Hahn, A. Mosedale, and B. Thornber. Large eddy simulation using high-resolution and high-order methods. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1899):2985, 2009.

[24] D. Drikakis and P.K. Smolarkiewicz. On spurious vortical structures. *Journal of Computational Physics*, 39(172):309–325, 2001.

[25] M. Hahn and D. Drikakis. Large eddy simulation of compressible turbulence using high-resolution methods. *International journal for numerical methods in fluids*, 47(8-9):971–977, 2005.